NLR

Dedicated to innovation in aerospace

Affine LPV Modeling: An H∞ Based Approach

Skander Taamallah [(1)], Xavier Bombois [(2)], and Paul M.J. Van den Hof [(3)]

[(1)] National Aerospace Laboratory (NLR) Amsterdam, and Delft Center for Systems and Control (DCSC), Delft University of Technology, Delft, The Netherlands
[(2)] Delft Center for Systems and Control (DCSC), Delft University of Technology, Delft, The Netherlands
[(3)] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

IEEE Conference on Decision and Control (CDC), 10-13 December 2013, Florence, Italy

Objective

- Given availability of:
  - high-fidelity, complex, **nonlinear model**

- Find a **reduced complexity model**:
  - suitable for high-performance **controller design**, over the complete operating regime
  - With open open-loop and closed-loop capabilities
  - retaining high computational efficiency, (e.g. on-line trajectory planning)

- Approach:
  - LPV model
  - Utilizing simulated input-output signal sequence, collected under the desired (non-stationary) operating conditions

Contribution

- Novel affine (quasi-) LPV modeling method:
  - applies to any numerically-based model
  - not restricted to equilibrium points, rather it captures the non-stationary and transient system behavior
  - not limited by practical implementation constraints such as state differentiation
  - With user-specified frequency-range of interest
  - Using the H_∞ norm paradigm

Problem Statement (1/4)

- From a Noise-free, Continuous-Time, Nonlinear Model given by

$$\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \qquad \mathbf{y}(t) = \mathbf{x}(t)$$

- Compute non-stationary linearizations, along a simulated (sampled) input-output signal sequence

$$\bar{A}_i = \left.\frac{\delta f(\mathbf{x}, \mathbf{u})}{\delta \mathbf{x}^T}\right|_{(\mathbf{x}_i, \mathbf{u}_i)} \qquad \bar{B}_i = \left.\frac{\delta f(\mathbf{x}, \mathbf{u})}{\delta \mathbf{u}^T}\right|_{(\mathbf{x}_i, \mathbf{u}_i)}$$

$$\mathbf{d}_i = f(\mathbf{x}_i, \mathbf{u}_i) - \bar{A}_i \mathbf{x}_i - \bar{B}_i \mathbf{u}_i$$

$$\mathcal{Z}_{Lin}^N := \{(\bar{A}_i, \bar{B}_i, \mathbf{d}_i)\}_{i=1}^N$$

Sequence of frozen LTI systems

- In order to obtain an affine LPV representation

$$\mathsf{S}(\theta(t)) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0 \mathbf{x}(t) + B_0 \mathbf{u}(t) + \ldots \\ \sum_{r=1}^{R} \theta_r(t)\big(A_r \mathbf{x}(t) + B_r \mathbf{u}(t)\big) \end{cases}$$

Problem Statement (2/4)

- First note the equivalence between model S(*θ(t)*)

$$\mathsf{S}(\theta(t)) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0\mathbf{x}(t) + B_0\mathbf{u}(t) + \dots \\ \sum_{r=1}^{R} \theta_r(t)\Big(A_r\mathbf{x}(t) + B_r\mathbf{u}(t)\Big) \end{cases}$$

- And model S(*η(t), ζ(t)*)

$$\mathsf{S}\big(\eta(t), \zeta(t)\big) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0\mathbf{x}(t) + B_0\mathbf{u}(t) + \dots \\ \sum_{s=1}^{S} \eta_s(t)\Big(L_s\mathbf{x}(t) + R_s\mathbf{u}(t)\Big) + \dots \\ \sum_{w=1}^{W} \zeta_w(t)\Big(T_w\mathbf{x}(t) + Z_w\mathbf{u}(t)\Big) \end{cases}$$

$$\mathcal{Z}_{Lin}^{N} := \{(\overline{A}_i, \overline{B}_i, \mathbf{d}_i)\}_{i=1}^{N}$$

$$\mathcal{Z}_{Lin}^{N} := \{(\overline{A}_i, \overline{B}_i, \mathbf{d}_i)\}_{i=1}^{N}$$

Problem Statement (3/4)

- Affine Model

$$\mathsf{S}\big(\eta(t),\zeta(t)\big) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0\mathbf{x}(t) + B_0\mathbf{u}(t) + \ldots \\ \sum_{s=1}^{S} \eta_s(t)\big(L_s\mathbf{x}(t) + R_s\mathbf{u}(t)\big) + \ldots \\ \sum_{w=1}^{W} \zeta_w(t)\big(T_w\mathbf{x}(t) + Z_w\mathbf{u}(t)\big) \end{cases}$$

- Next, we need to identify
- affine term (A_0,B_0)
- basis functions (L_s,R_s) and (T_w,Z_w)
- scheduling parameters $\eta_s\,(t)$ *and* $\zeta_w\,(t)$

Problem Statement (4/4)

- Affine Model

$$\mathsf{S}\big(\eta(t),\zeta(t)\big) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0\mathbf{x}(t) + B_0\mathbf{u}(t) + \ldots \\ \sum_{s=1}^{S} \eta_s(t)\big(L_s\mathbf{x}(t) + R_s\mathbf{u}(t)\big) + \ldots \\ \sum_{w=1}^{W} \zeta_w(t)\big(T_w\mathbf{x}(t) + Z_w\mathbf{u}(t)\big) \end{cases}$$

- From

$$\mathcal{Z}_{Lin}^{N} := \{(\bar{A}_i, \bar{B}_i, \mathbf{d}_i)\}_{i=1}^{N}$$

- **Problem I**: The information in $(\bar{A}_i, \bar{B}_i)$ is used to identify
- affine term (A_0, B_0), basis functions (L_s, R_s)
- scheduling parameters $\eta_s(t)$

- **Problem II**: The information in $(\mathbf{d}_i)$ is used to identify
- basis functions (T_w, Z_w)
- scheduling parameters $\zeta_w(t)$

- **Problem III:** find static mappings from $(\mathbf{x}(t),\mathbf{u}(t))$ to the scheduling parameters $\eta_s(t)$ *and* $\zeta_w(t)$

Defining The Optimization Problems (1/2)

- We define $\bar{G}_i(s) := \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline \bar{C} & \bar{D} \end{array}\right]$ $\quad \bar{C} = I,\ \bar{D} = 0$
- And

$$G_i(s) \ := \left[\begin{array}{c|c} A_0 + \sum_{s=1}^{S} \eta_{s_i} L_s & B_0 + \sum_{s=1}^{S} \eta_{s_i} R_s \\ \hline C & D \end{array}\right] \qquad C = I,\ D = 0$$

- Then **Problem I**

$$(\hat{A}_0, \hat{B}_0, \hat{L}, \hat{R}, \hat{\Pi}) = \underset{\substack{\left(A_0 \in \mathbb{R}^{n_x \times n_x}, B_0 \in \mathbb{R}^{n_x \times n_u}\right) \\ \left(L_s \in \mathbb{R}^{n_x \times n_x}, R_s \in \mathbb{R}^{n_x \times n_u}, \eta_{s_i} \in \mathbb{R}\right)}}{\arg\inf} J_1$$

$$J_1 := \sum_{i=1}^{N} \left\| \bar{G}_i(s) - G_i(s) \right\|_{\Delta_\omega}$$

- With $\hat{L} := [\hat{L}_s]_{S n_x \times n_x}$, $\hat{R} := [\hat{R}_s]_{S n_x \times n_u}$, and $\hat{\Pi} := [\hat{\eta}_{s_i}]_{S \times N}$
- And $\|G\|_{\Delta_\omega} := \sup_{\omega \in \Delta_\omega} \bar{\sigma}(G(j\omega))$ from KYP Lemma (spectral constraints)

Defining The Optimization Problems (2/2)

- **Problem II**

$$(\hat{T}, \hat{Z}, \hat{\Xi}) = \underset{\left(T_w \in \mathbb{R}^{n_x \times n_x}, Z_w \in \mathbb{R}^{n_x \times n_u}, \zeta_{w_i} \in \mathbb{R}\right)}{\arg\inf} J_2$$

$$J_2 := \sum_{i=1}^{N} \left\| \mathbf{d}_i - \sum_{w=1}^{W} \zeta_{w_i} \left(T_w \mathbf{x}_i + Z_w \mathbf{u}_i \right) \right\|_2$$

- With $\hat{T} := [\hat{T}_w]_{Wn_x \times n_x}$, $\hat{Z} := [\hat{Z}_w]_{Wn_x \times n_u}$, and $\hat{\Xi} := [\hat{\zeta}_{w_i}]_{W \times N}$

A Solution to Problem I

- First obtain the affine term

$$(\hat{A}_0, \hat{B}_0) = \arg \inf_{\left(A_0 \in \mathbb{R}^{n_x \times n_x}, B_0 \in \mathbb{R}^{n_x \times n_u}\right)} \sum_{i=1}^{N} \|\bar{G}_i(s) - G_0(s)\|_{\Delta_\omega}$$

Restricts the search space to $\mathcal{Z}^N_{Lin}$, i.e. one of the frozen models

- Next obtain the basis functions $\hat{L} := [\hat{L}_s]_{S\, n_x \times n_x},\ \hat{R} := [\hat{R}_s]_{S\, n_x \times n_u}$ based upon the use of SVD decompositions

- Finally obtain the scheduling parameters: two-stage modus operandi

  - an initialization stage
  - a nonlinear refinement stage

$$\forall i \in \{1, \ldots, N\} \quad [\hat{\eta}_{s_i}]_{S \times 1} = \cdots$$
$$\arg \inf_{\left(\eta_{s_i} \in \mathbb{R}\right)} \|\bar{G}_i(s) - G_i(s)\|_{\Delta_\omega}$$

A Solution to Problem II

- Suppose we can find scheduling parameters $\zeta_w(t)$ *and* basis functions (T_w, Z_w) such that

$$\forall i \in \{1, \ldots, N\} \quad \mathbf{d}_i \begin{bmatrix} \mathbf{x}_i \\ \mathbf{u}_i \end{bmatrix}^{\dagger} \approx \begin{bmatrix} \sum_{w=1}^{W} \zeta_{w_i} T_w & \sum_{w=1}^{W} \zeta_{w_i} Z_w \end{bmatrix}$$

- With $[\cdot]^{\dagger}$ the left inverse, then by right-multiplying both sides with $[\mathbf{x}_i^T \ \ \mathbf{u}_i^T]^T$ we recover

$$\mathbf{d}_i \approx \sum_{w=1}^{W} \zeta_{w_i} \left(T_w \mathbf{x}_i + Z_w \mathbf{u}_i \right)$$

- Solution based upon SVD decompositions and least-squares

Problem III

- We aim to find suitable representations

$$\eta(t) = g(\mathbf{x}(t), \mathbf{u}(t)) \qquad \zeta(t) = h(\mathbf{x}(t), \mathbf{u}(t))$$

- By illustrating the applicability of a standard two-layer feedforward neural networks
- The first layer being sigmoid and the second linear, with a hyperbolic tangent activation transfer function in the hidden layer, and backpropagation training for the weights and biases

Simulation Results (1/5)

- Pointmass pendulum, the rotational motion is given by

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -bx_2(t) - \beta^2 \sin x_1(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha(u(t)) \end{bmatrix}$$

- With a fictional nonlinearity $\alpha(u(t)) := u(t)^2 \sin u(t)$
- And states $[x_1 \; x_2]^T = [\theta \; \dot{\theta}]^T$
- To derive the LPV model, we excite the pendulum, from its rest position, with a 10 sec long sine-sweep
- Sine-sweep defined as: $u(t) = A \sin(2\pi. f .t)$
  $A = 2$, $f \in [0.05-1]$ Hz, sampled with a period $T = 0.05$ sec, resulting in 201 data points

Simulation Results (2/5)

- The optimal affine term $\hat{G}_0(s) := \left[\begin{array}{c|c} \hat{A}_0 & \hat{B}_0 \\ \hline C & D \end{array}\right]$ corresponds to the LTI model for *i=59*
- We retain all basis functions in Problem I & II, i.e. in this simulation case S=2, W=3
- We use a 5-neurons neural network

Simulation Results (3/5)

- Next, we use fresh data sets, namely sine-inputs at various amplitudes and frequencies, and compare time-domain outputs of the original nonlinear model (NM) with the LPV model

- To compare results we use the Best-FiT (BFT) and Variance-Accounted-For (VAF) metrics

Simulation Results (4/5)

- Validation results: given in Table, Left value is BFT (%), Right value is VAF (%)

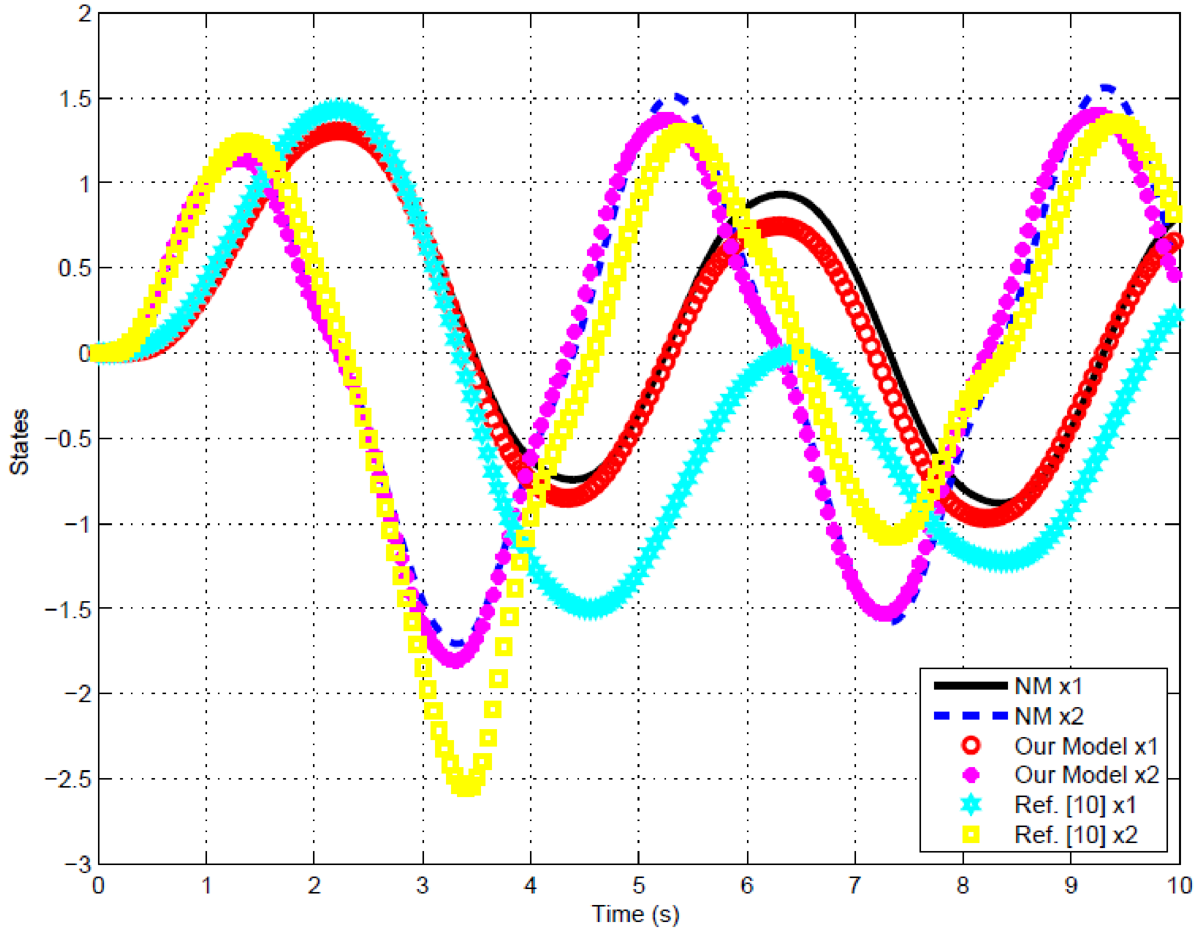
Input Amplitude	Input Frequency (Hz) 0.25	0.5	0.75	1
0.5	65 95	48 93	37 91	37 87
1	74 96	71 95	53 95	35 92
1.5	86 98	82 98	68 97	39 94
2	86 99	81 99	92 100	85 99

- LPV model exhibits very good to excellent fit with the nonlinear model (NM), for an input amplitude equal to the value used during estimation (Amplitude $A = 2$)

- Naturally, LPV model accuracy diminishes as input amplitude is moved away from value used during estimation

Simulation Results (5/5)

- For illustration, case (A = 2, f = 0.25 Hz) is shown



Conclusion

- Novel and comprehensive affine quasi-LPV modeling method
- For high model fidelity in open-loop, one could keep a maximum number of basis functions
- Whereas for controller design, one could potentially cope with fewer basis functions
- Scheduling parameters: static dependence only.
- Experiment design to be developed